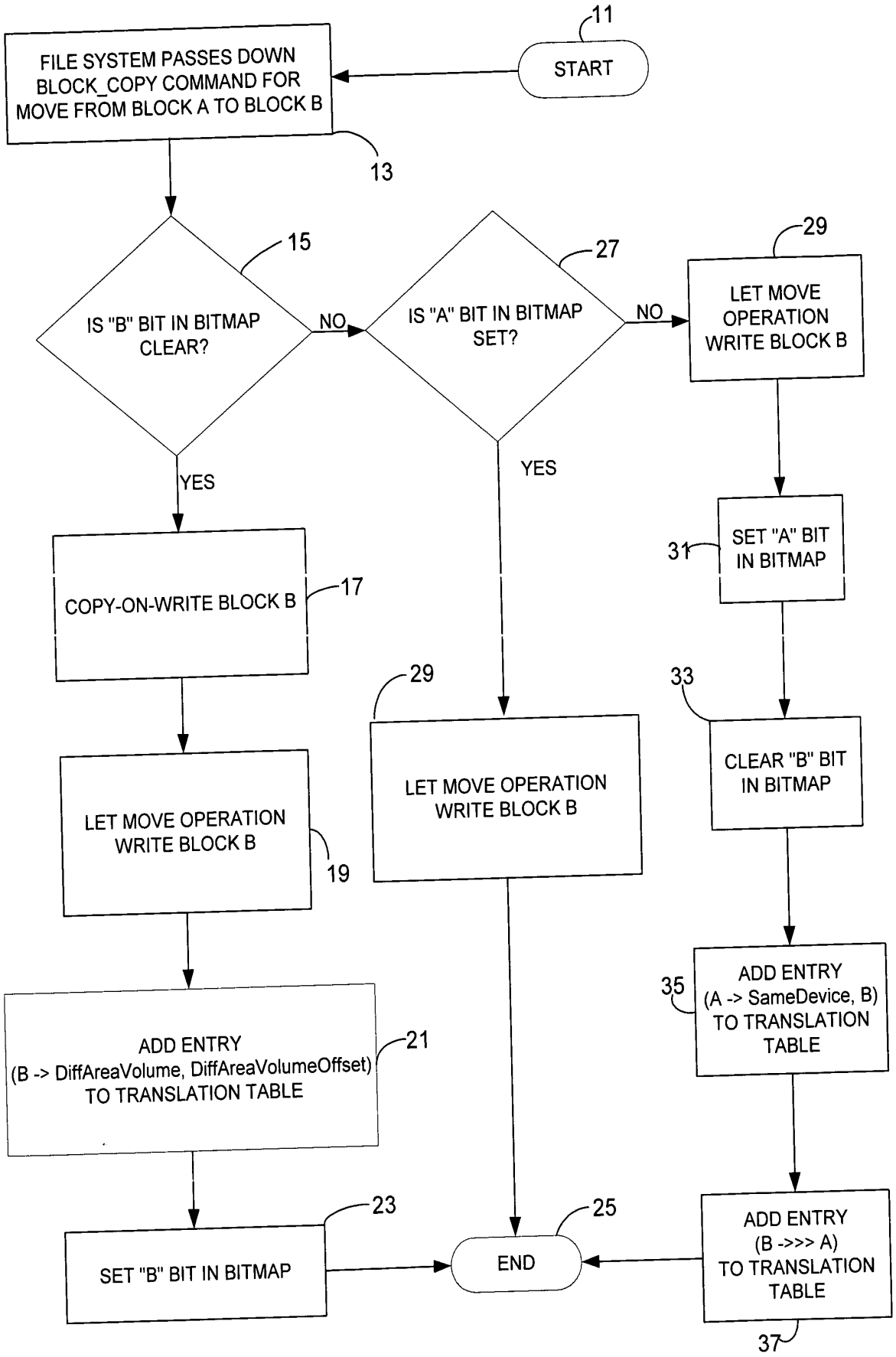

11
START
FILE SYSTEM PASSES DOWN BLOCK_COPY COMMAND FOR MOVE FROM BLOCK A TO BLOCK B
13
15
IS "B" BIT IN BITMAP CLEAR?
NO
27
IS "A" BIT IN BITMAP SET?
NO
29
LET MOVE OPERATION WRITE BLOCK B
YES
YES
31
SET "A" BIT IN BITMAP
COPY-ON-WRITE BLOCK B
17
29
LET MOVE OPERATION WRITE BLOCK B
33
CLEAR "B" BIT IN BITMAP
LET MOVE OPERATION WRITE BLOCK B
19
35
ADD ENTRY (A -> SameDevice, B) TO TRANSLATION TABLE
ADD ENTRY (B -> DiffAreaVolume, DiffAreaVolumeOffset) TO TRANSLATION TABLE
21
23
25
SET "B" BIT IN BITMAP
END
ADD ENTRY (B ->>> A) TO TRANSLATION TABLE
37

FIG. 1

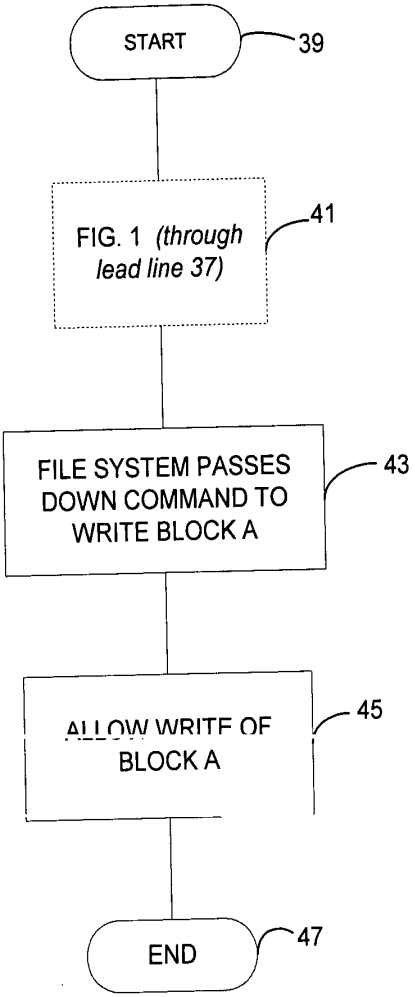


FIG. 2A

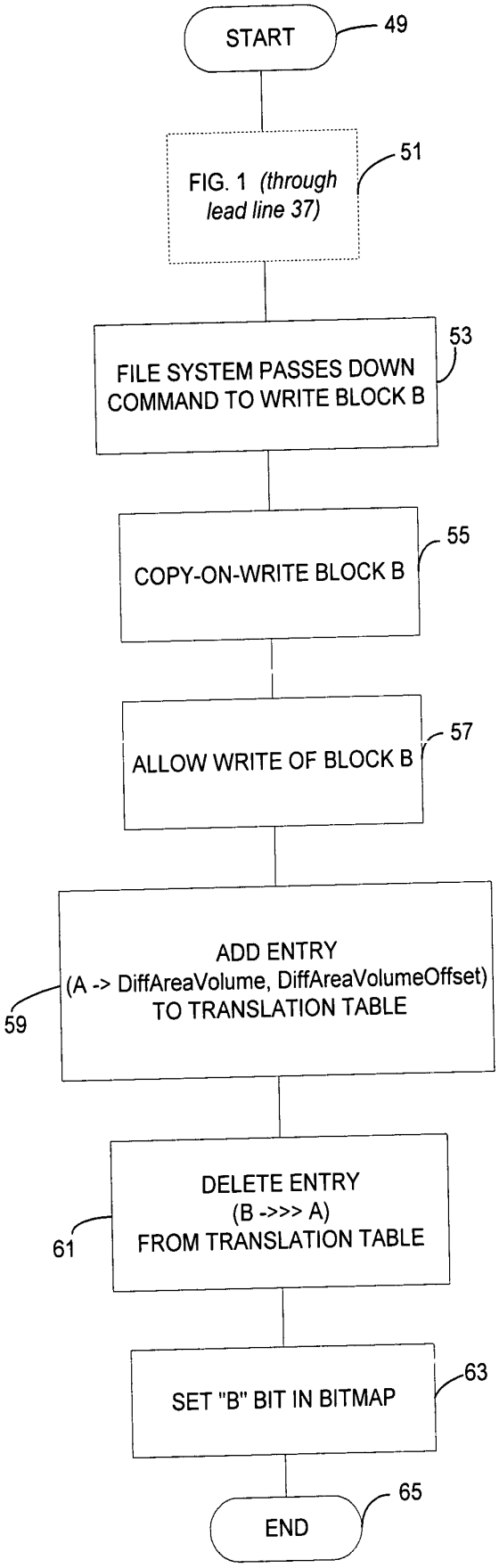


FIG. 2B

FIG. 2

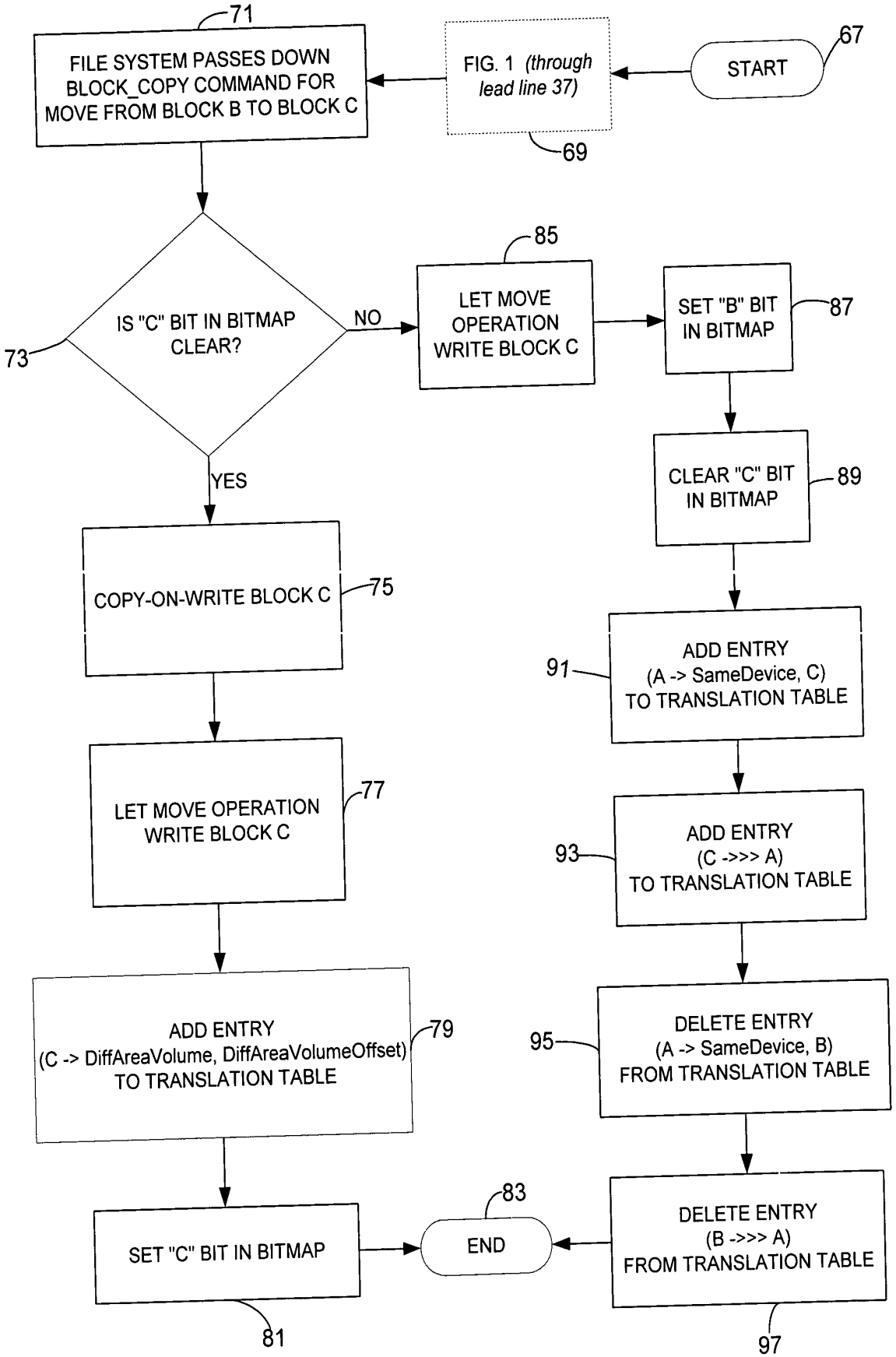

START
67
FIG. 1 (through lead line 37)
69
FILE SYSTEM PASSES DOWN BLOCK_COPY COMMAND FOR MOVE FROM BLOCK B TO BLOCK C
71
IS "C" BIT IN BITMAP CLEAR?
73
NO
YES
LET MOVE OPERATION WRITE BLOCK C
85
SET "B" BIT IN BITMAP
87
CLEAR "C" BIT IN BITMAP
89
COPY-ON-WRITE BLOCK C
75
ADD ENTRY (A -> SameDevice, C) TO TRANSLATION TABLE
91
LET MOVE OPERATION WRITE BLOCK C
77
ADD ENTRY (C ->>> A) TO TRANSLATION TABLE
93
ADD ENTRY (C -> DiffAreaVolume, DiffAreaVolumeOffset) TO TRANSLATION TABLE
79
DELETE ENTRY (A -> SameDevice, B) FROM TRANSLATION TABLE
95
SET "C" BIT IN BITMAP
81
END
83
DELETE ENTRY (B ->>> A) FROM TRANSLATION TABLE
97

FIG. 3

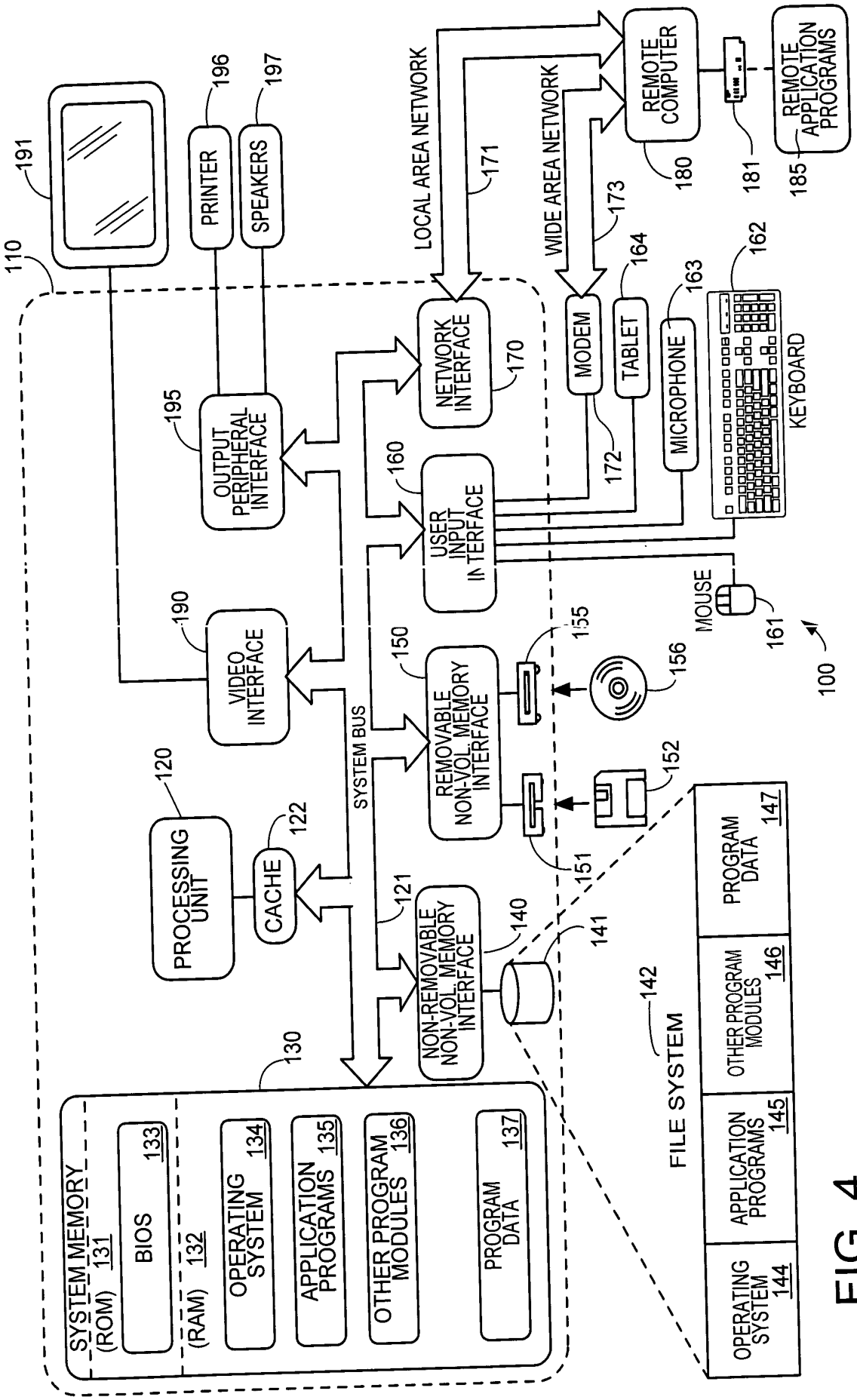

110
191
PRINTER
196
SPEAKERS
197
LOCAL AREA NETWORK
171
WIDE AREA NETWORK
173
REMOTE COMPUTER
180
181
REMOTE APPLICATION PROGRAMS
185
NETWORK INTERFACE
170
MODEM
172
TABLET
164
MICROPHONE
163
KEYBOARD
162
OUTPUT PERIPHERAL INTERFACE
195
160
USER INPUT INTERFACE
MOUSE
161
100
VIDEO INTERFACE
190
150
155
156
SYSTEM BUS
REMOVABLE NON-VOL. MEMORY INTERFACE
120
PROCESSING UNIT
CACHE
122
121
151
152
140
141
NON-REMOVABLE NON-VOL. MEMORY INTERFACE
130
SYSTEM MEMORY
(ROM) 131
BIOS 133
(RAM) 132
OPERATING SYSTEM 134
APPLICATION PROGRAMS 135
OTHER PROGRAM MODULES 136
PROGRAM DATA 137
142
FILE SYSTEM
OPERATING SYSTEM 144
APPLICATION PROGRAMS 145
OTHER PROGRAM MODULES 146
PROGRAM DATA 147

FIG. 4

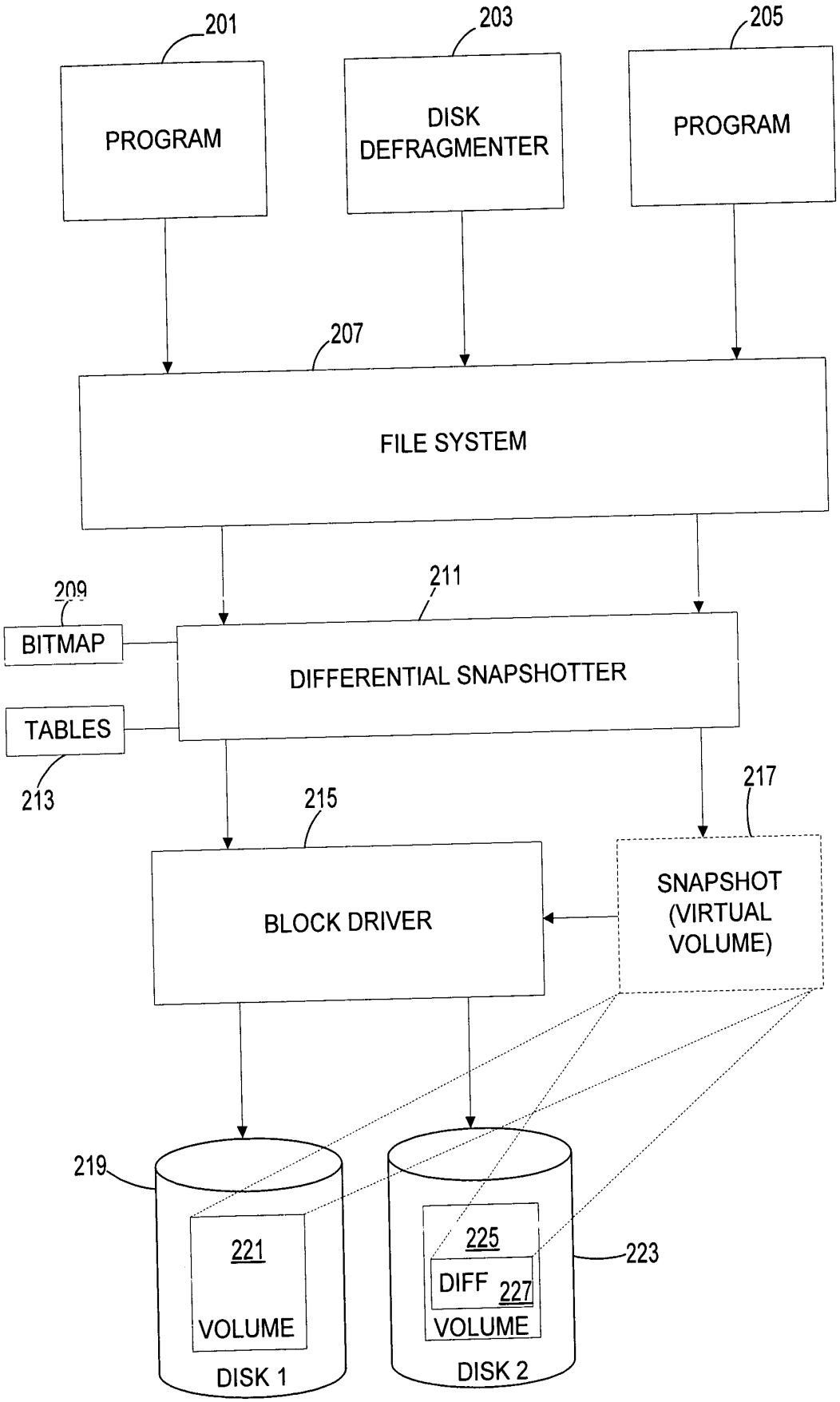

201
203
205
PROGRAM
DISK DEFRAGMENTER
PROGRAM
207
FILE SYSTEM
209
211
BITMAP
DIFFERENTIAL SNAPSHOTTER
TABLES
213
215
217
BLOCK DRIVER
SNAPSHOT (VIRTUAL VOLUME)
219
221
VOLUME
DISK 1
225
DIFF 227
VOLUME
DISK 2
223

FIG. 5

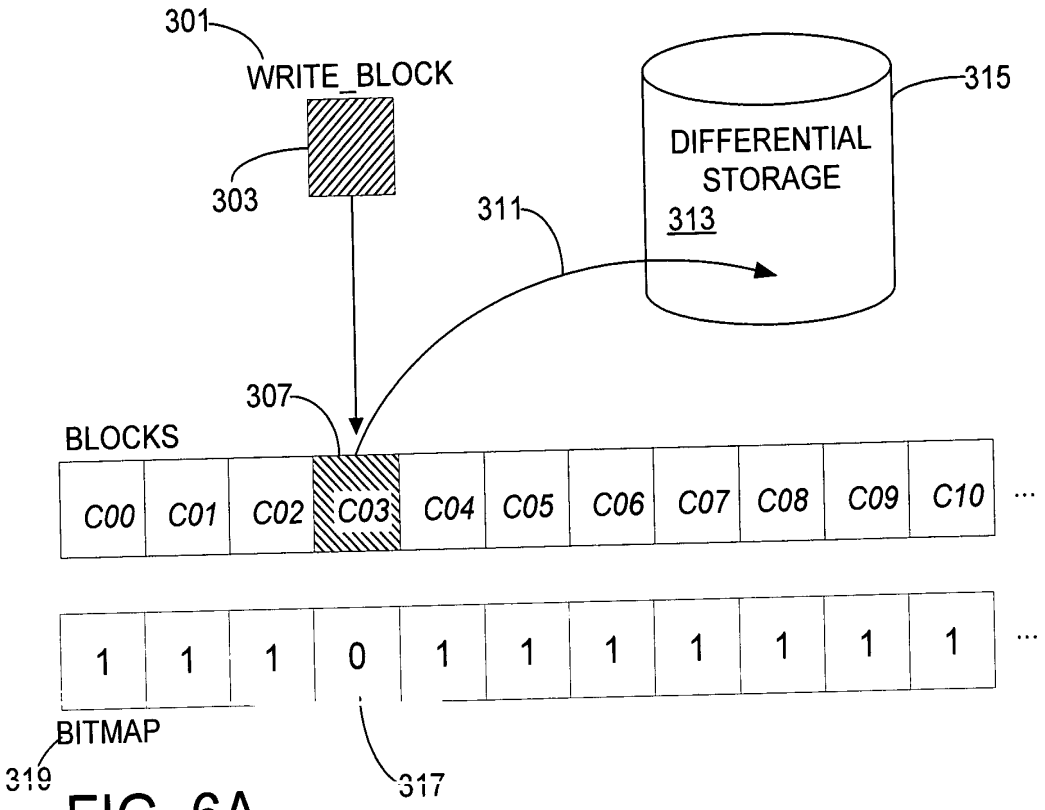

301
WRITE_BLOCK
303
311
DIFFERENTIAL STORAGE
313
315
307
BLOCKS
C00 C01 C02 C03 C04 C05 C06 C07 C08 C09 C10 ...
1 1 1 0 1 1 1 1 1 1 1 ...
BITMAP
319
317

FIG. 6A

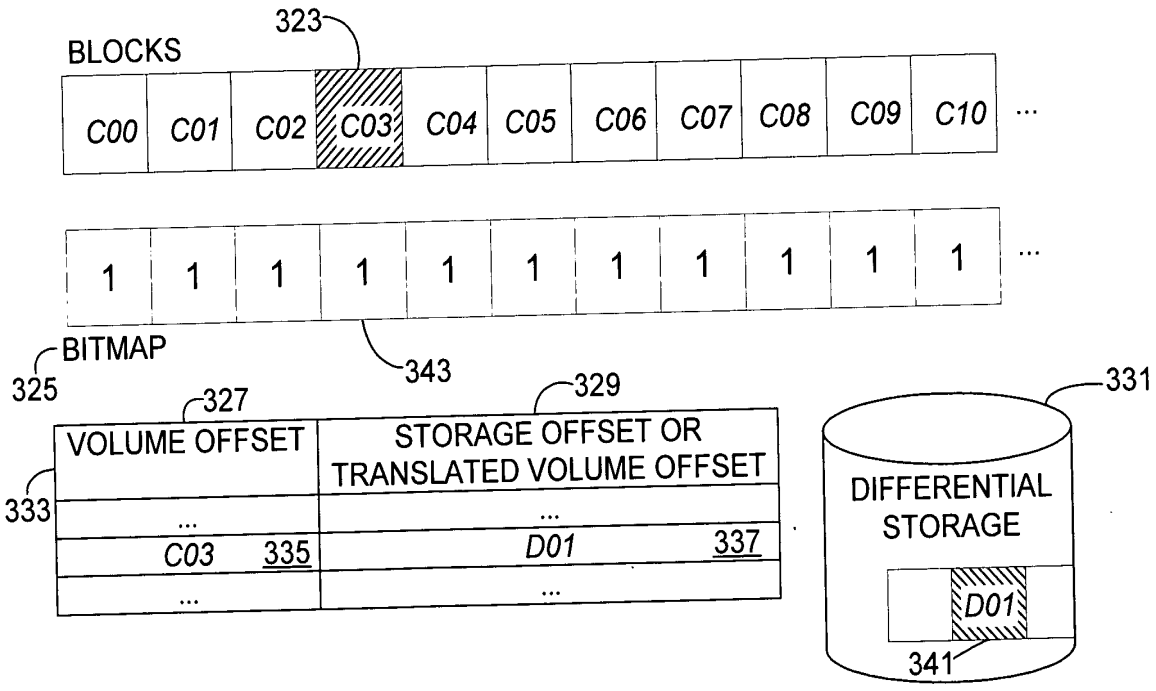

323
BLOCKS
C00 C01 C02 C03 C04 C05 C06 C07 C08 C09 C10 ...
1 1 1 1 1 1 1 1 1 1 1 ...
BITMAP
325
343
327
329
331
VOLUME OFFSET
STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
333
...
C03 335
D01 337
DIFFERENTIAL STORAGE
D01
341

FIG. 6B

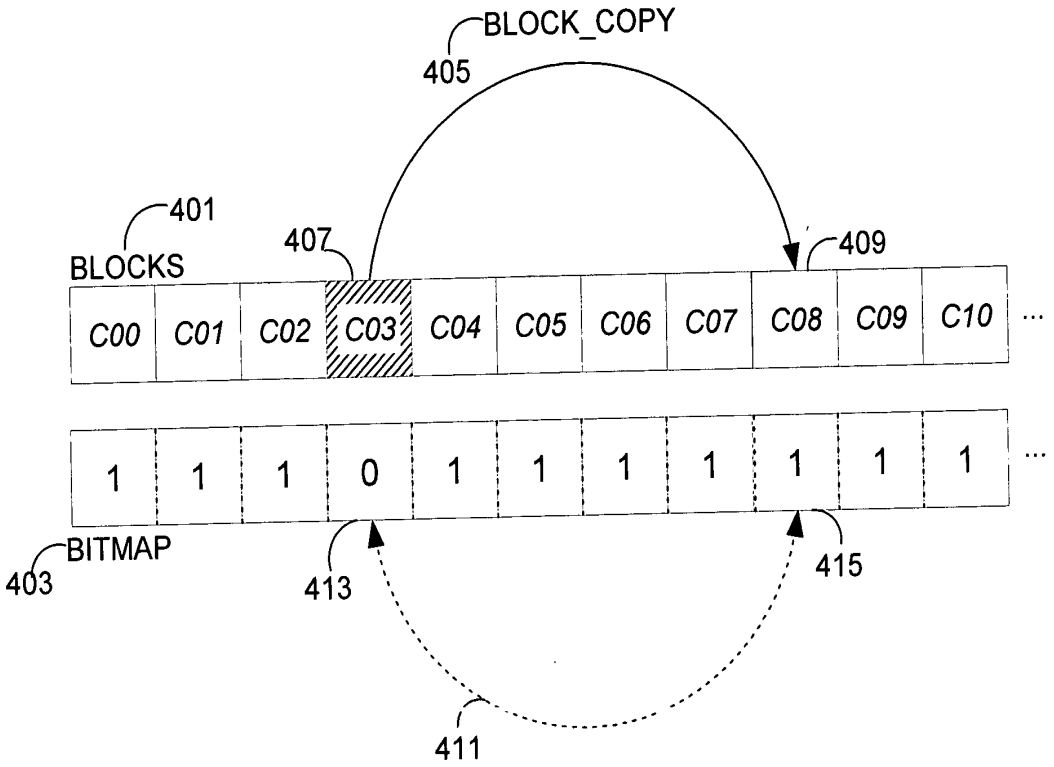

BLOCK_COPY
405
401
BLOCKS
407
409
C00
C01
C02
C03
C04
C05
C06
C07
C08
C09
C10
1
1
1
0
1
1
1
1
1
1
1
BITMAP
403
413
415
411

FIG. 7A

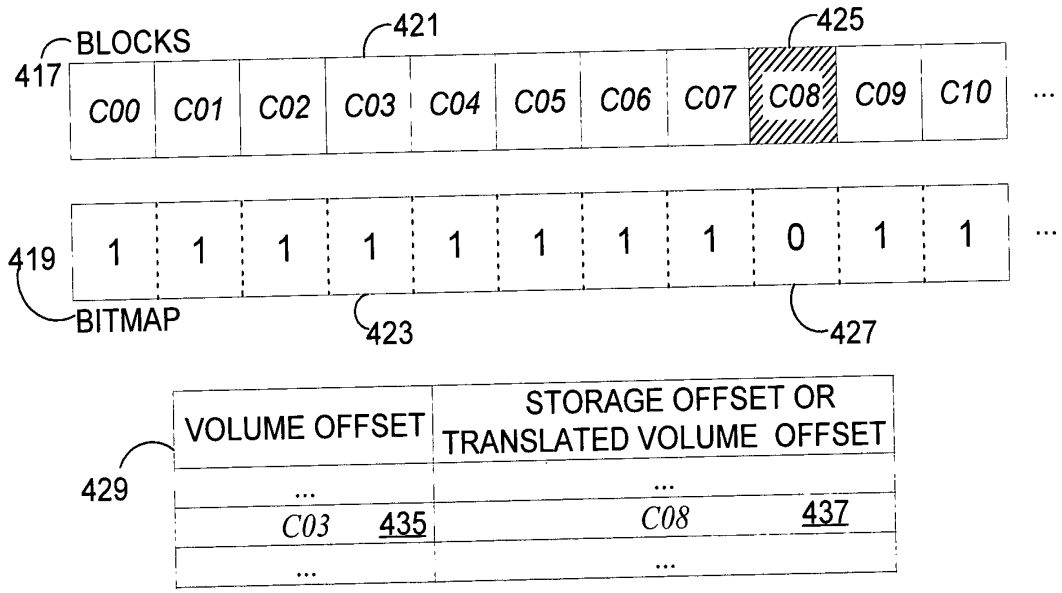

BLOCKS
417
421
425
C00
C01
C02
C03
C04
C05
C06
C07
C08
C09
C10
1
1
1
1
1
1
1
1
0
1
1
419
BITMAP
423
427
VOLUME OFFSET
STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
429
C03
435
C08
437

FIG. 7B

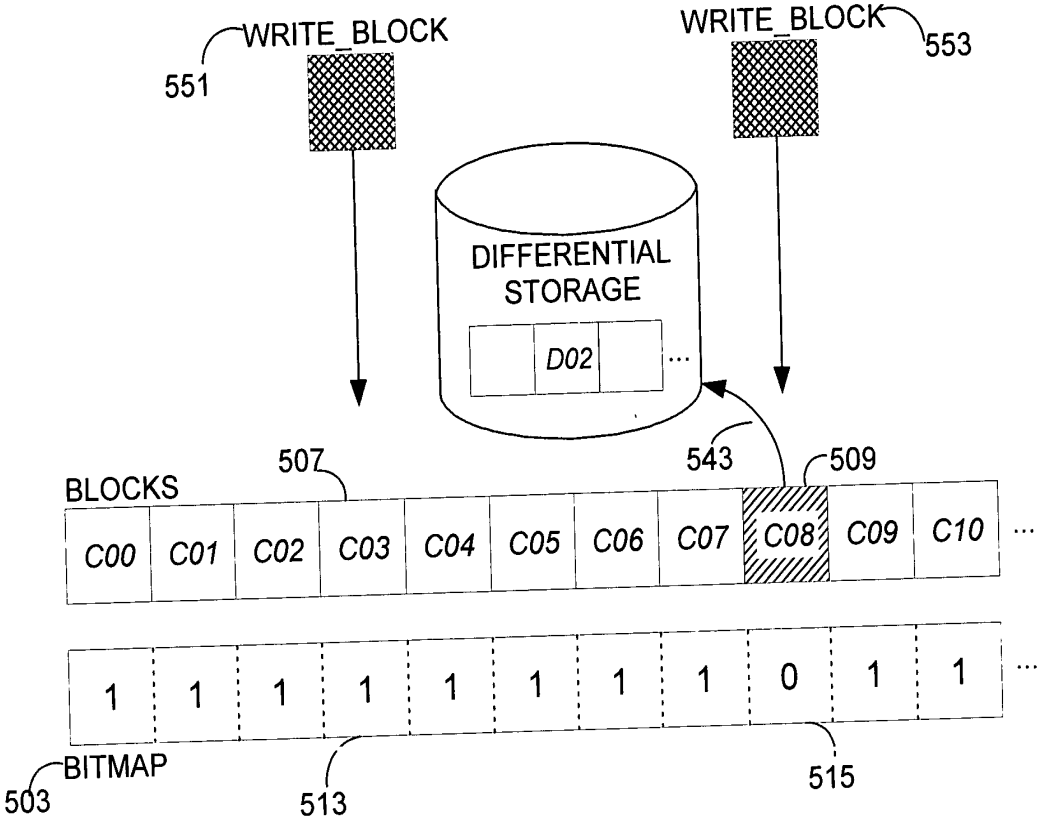

WRITE_BLOCK
551
WRITE_BLOCK
553
DIFFERENTIAL STORAGE
D02
543
507
509
BLOCKS
C00 C01 C02 C03 C04 C05 C06 C07 C08 C09 C10
1 1 1 1 1 1 1 1 0 1 1
BITMAP
503
513
515

FIG. 8A

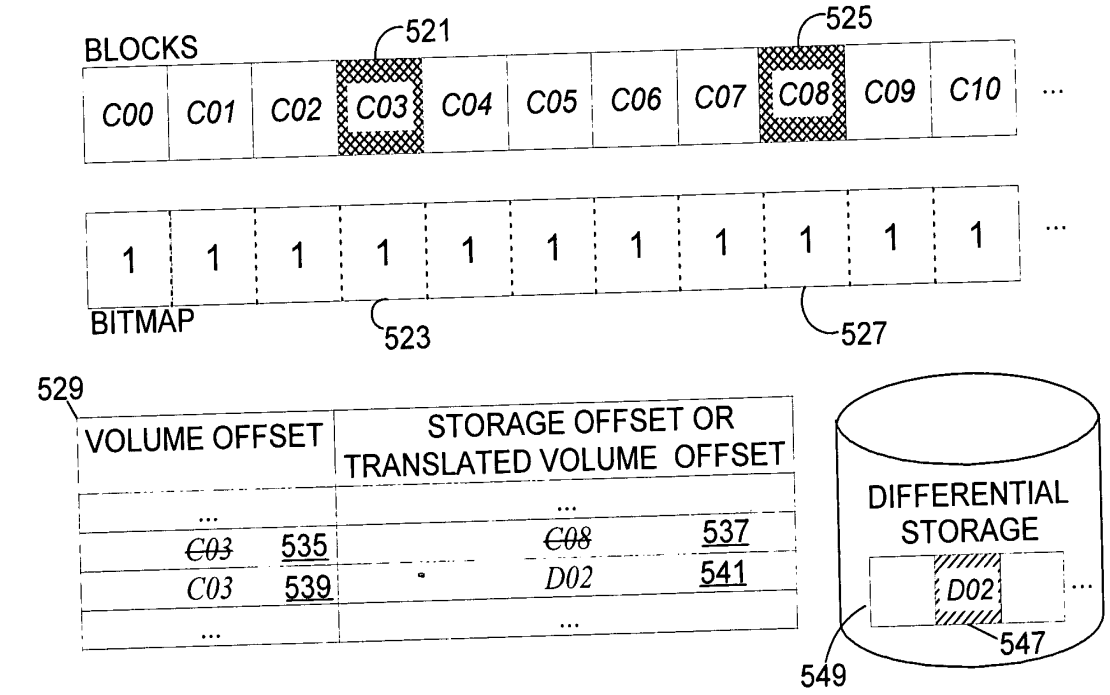
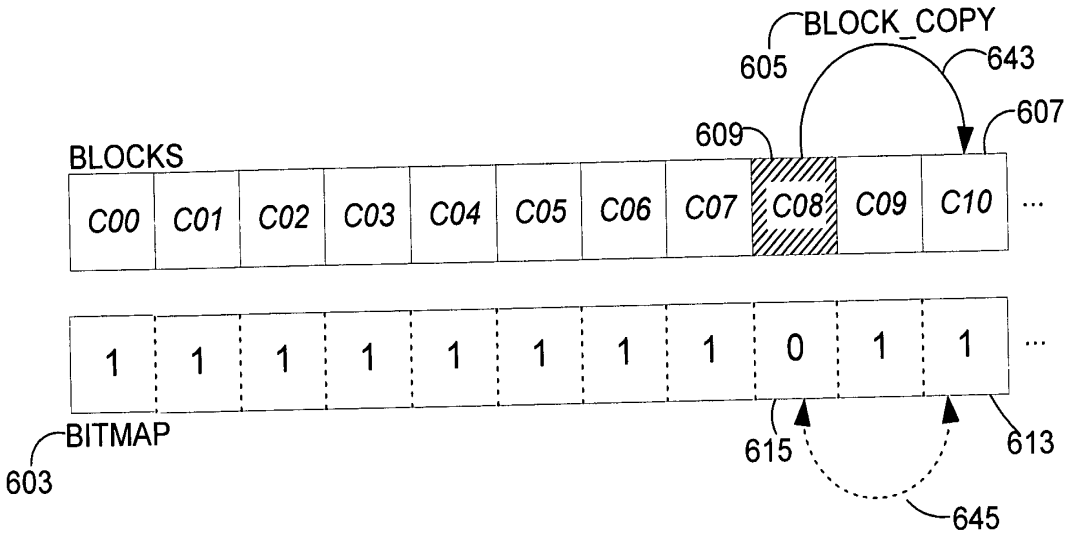

BLOCKS
521
525
C00 C01 C02 C03 C04 C05 C06 C07 C08 C09 C10
1 1 1 1 1 1 1 1 1 1 1
BITMAP
523
527
529
VOLUME OFFSET
STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
C03 535
C03 539
C08 537
D02 541
DIFFERENTIAL STORAGE
D02
547
549

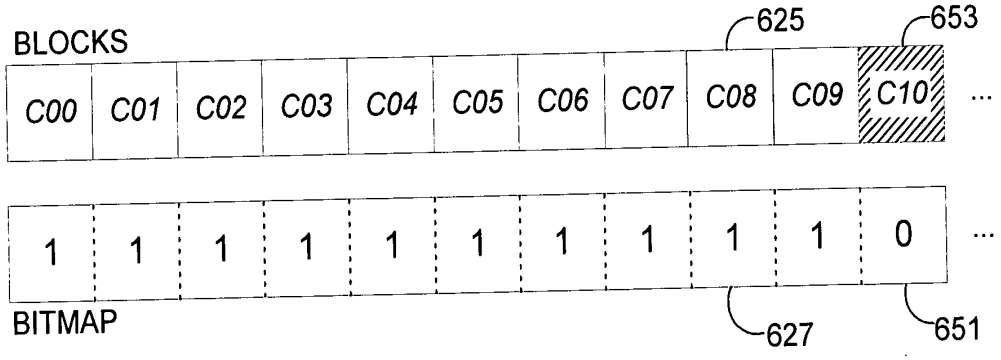
FIG. 8B



VOLUME OFFSET	STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
...	...
C03 635	C08 637
...	

629

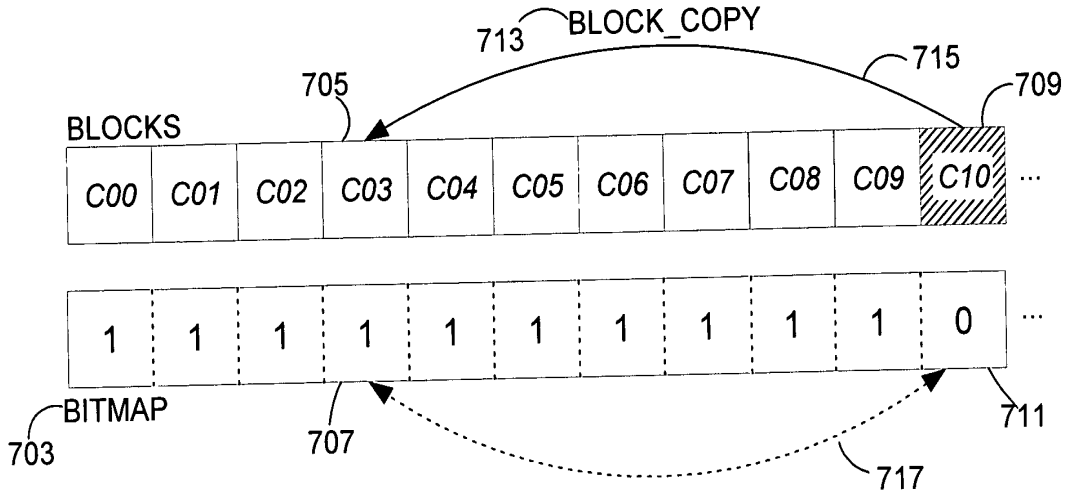
FIG. 9A



VOLUME OFFSET	STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
...	...
~~C03~~	~~C08~~ 641
C03 647	C10 649
...	...

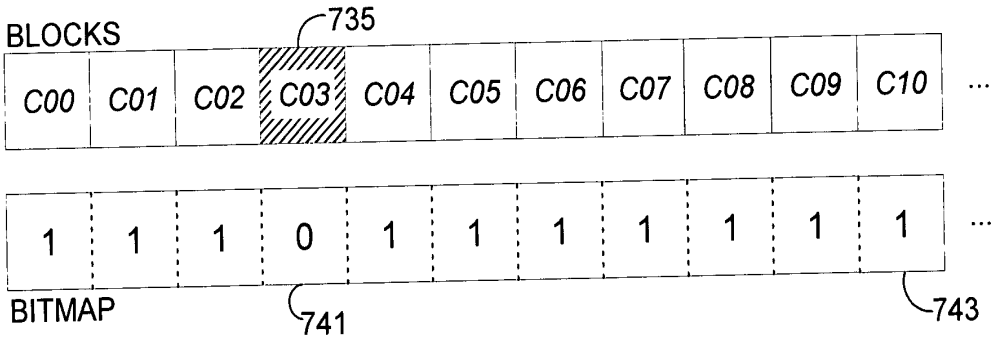
655

FIG. 9B



719	VOLUME OFFSET	STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
	...	...
	~~C03~~	~~C08~~
	~~C03~~ 721	~~C10~~ 731
	...	...

FIG. 10A



745	VOLUME OFFSET	749 STORAGE OFFSET OR TRANSLATED VOLUME OFFSET
	...	...
	~~C03~~	~~C08~~
	~~C03~~	~~C10~~
	~~C03~~ 755	~~C03~~ 761
	...	...

FIG. 10B

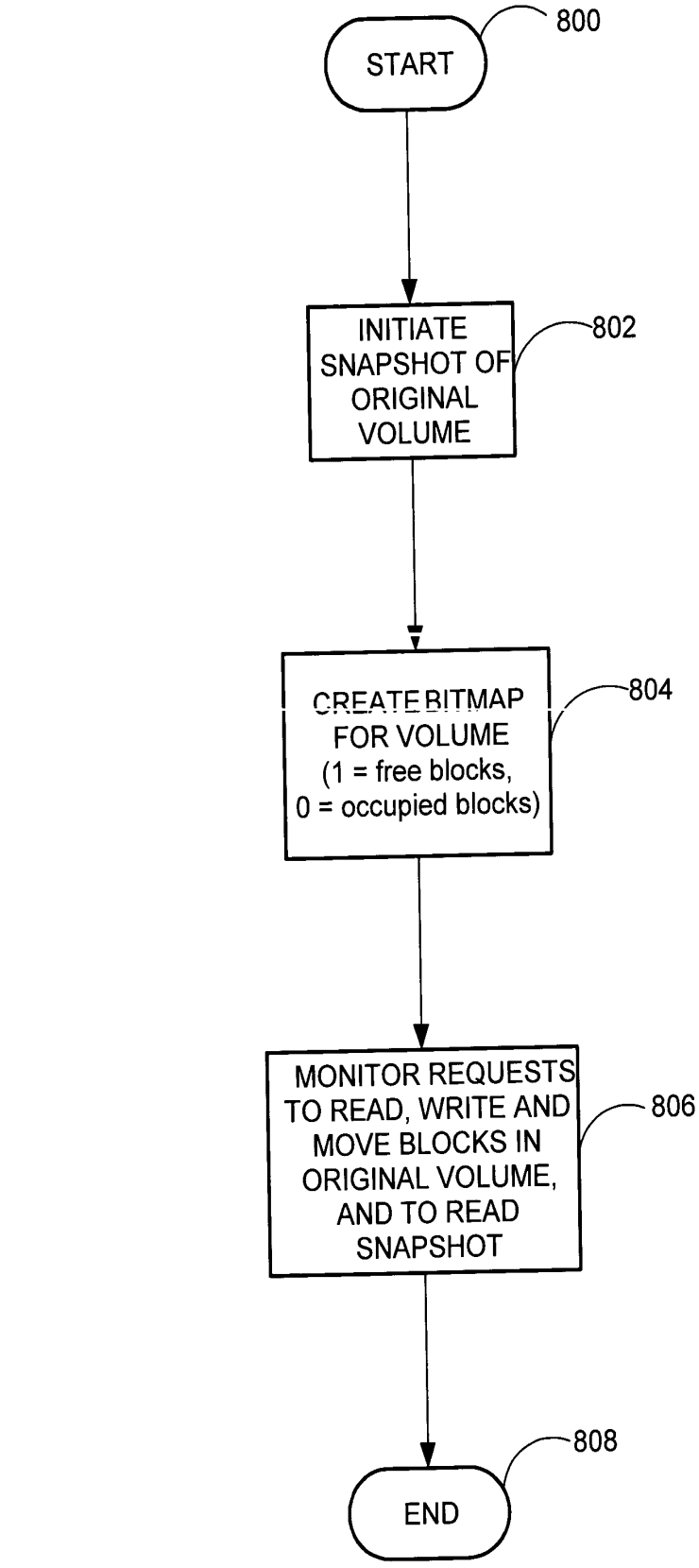

800
START
INITIATE SNAPSHOT OF ORIGINAL VOLUME
802
CREATE BITMAP FOR VOLUME (1 = free blocks, 0 = occupied blocks)
804
MONITOR REQUESTS TO READ, WRITE AND MOVE BLOCKS IN ORIGINAL VOLUME, AND TO READ SNAPSHOT
806
808
END

FIG. 11

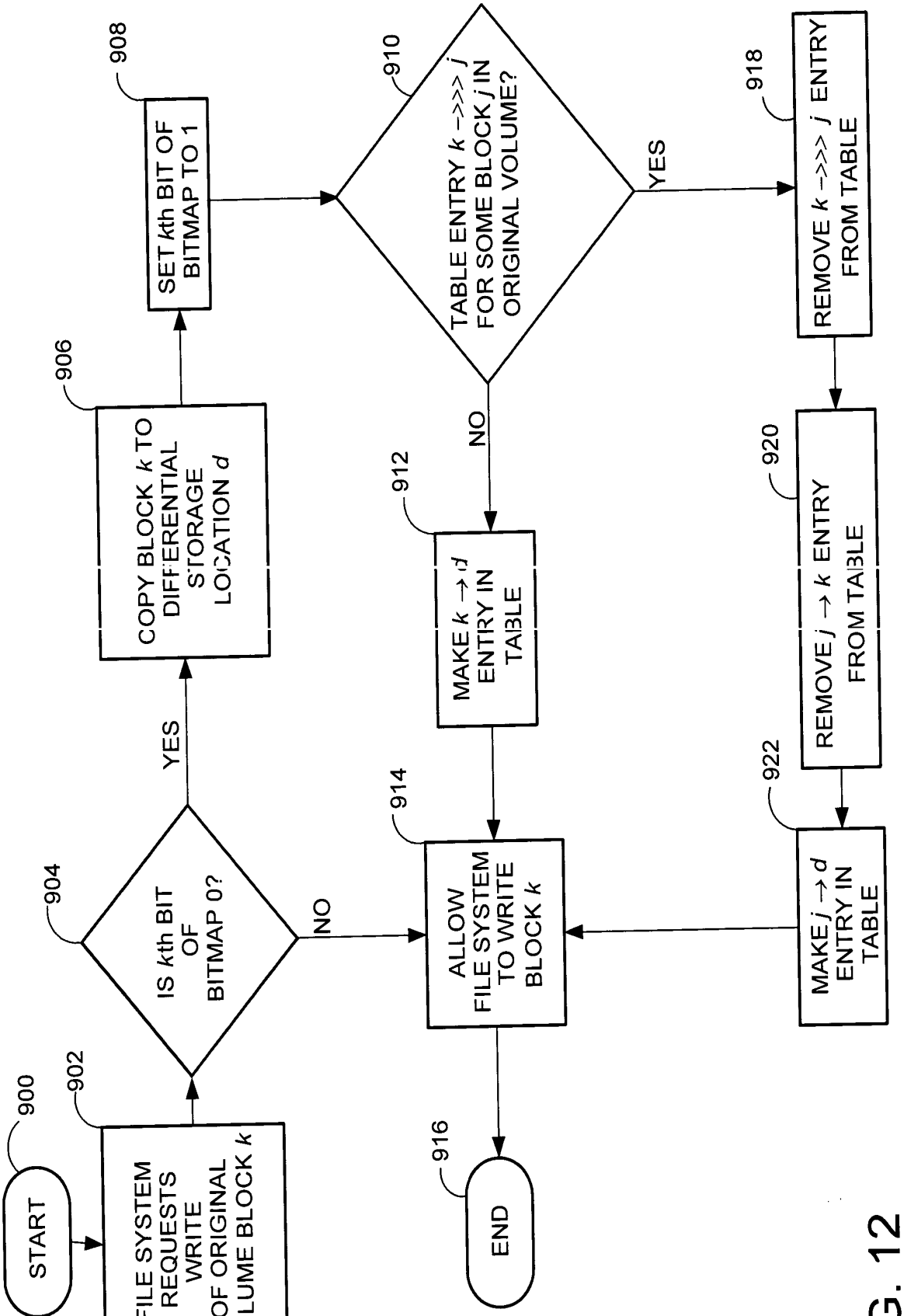

900
START
902
FILE SYSTEM REQUESTS WRITE OF ORIGINAL VOLUME BLOCK k
904
IS kth BIT OF BITMAP 0?
YES
NO
906
COPY BLOCK k TO DIFFERENTIAL STORAGE LOCATION d
908
SET kth BIT OF BITMAP TO 1
910
TABLE ENTRY k ->>> j FOR SOME BLOCK j IN ORIGINAL VOLUME?
YES
NO
912
MAKE k → d ENTRY IN TABLE
914
ALLOW FILE SYSTEM TO WRITE BLOCK k
916
END
918
REMOVE k ->>> j ENTRY FROM TABLE
920
REMOVE j → k ENTRY FROM TABLE
922
MAKE j → d ENTRY IN TABLE

FIG. 12

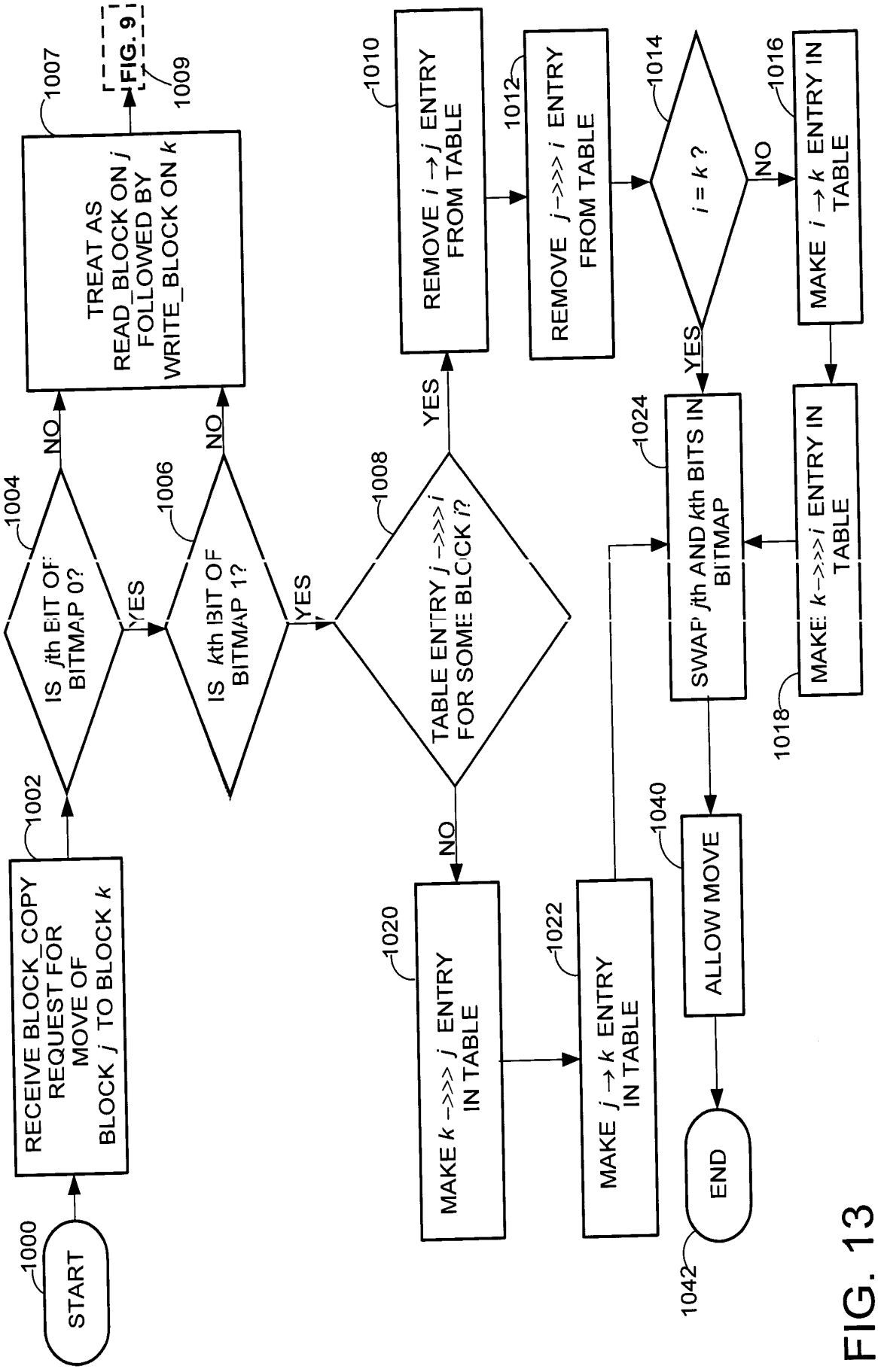

1000
START
1002
RECEIVE BLOCK_COPY REQUEST FOR MOVE OF BLOCK j TO BLOCK k
1004
IS jth BIT OF BITMAP 0?
NO
YES
1006
IS kth BIT OF BITMAP 1?
NO
YES
1007
TREAT AS READ_BLOCK ON j FOLLOWED BY WRITE_BLOCK ON k
1009
FIG. 9
1008
TABLE ENTRY j ->>> i FOR SOME BLOCK i?
YES
NO
1010
REMOVE i → j ENTRY FROM TABLE
1012
REMOVE j ->>> i ENTRY FROM TABLE
1014
i = k ?
YES
NO
1016
MAKE i → k ENTRY IN TABLE
1018
MAKE: k ->>> i ENTRY IN TABLE
1020
MAKE k ->>> j ENTRY IN TABLE
1022
MAKE j → k ENTRY IN TABLE
1024
SWAP jth AND kth BITS IN BITMAP
1040
ALLOW MOVE
1042
END

FIG. 13

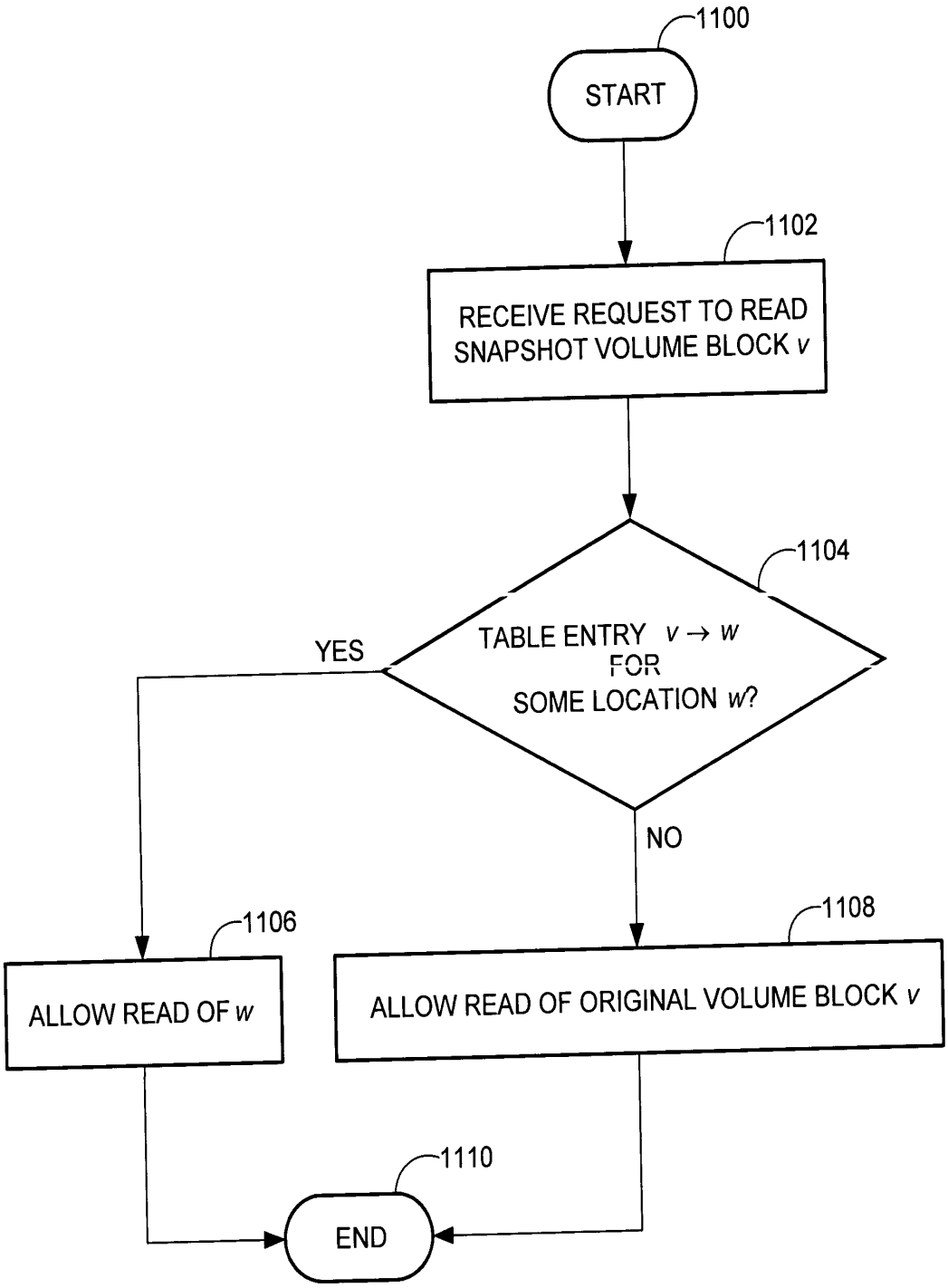

1100
START
1102
RECEIVE REQUEST TO READ SNAPSHOT VOLUME BLOCK v
1104
TABLE ENTRY v → w FOR SOME LOCATION w?
YES
NO
1106
ALLOW READ OF w
1108
ALLOW READ OF ORIGINAL VOLUME BLOCK v
1110
END

FIG. 14